```java
package semaphore.trafficLight.simple;

import java.awt.Dimension;

public class SimpleTrafficLight implements Paintable, TrafficLight {

    private Point position = new Point(0, 0);
    private Dimension dimension = new Dimension(70, 180);

    private Image mask;
    private SpotLight yellow;
    private SpotLight green;
    private SpotLight red;

    private String currentRelativePath()
    {
        return "/" +
        this.getClass()
        .getPackageName()
        .toString()
        .replace('.', '/') +
        "/";
    }

    private SpotLight createSpot (String color) throws IOException
    {
        final String path = currentRelativePath() + "img/";
        URL url;

        url = getClass().getResource(path + color + "On.png");
        Image maskOn = ImageIO.read(url);

        url = getClass().getResource(path + color + "Off.png");
        Image maskOff = ImageIO.read(url);

        SpotLight spot = new SpotLight(maskOn, maskOff);

        return spot;
    }

    private void create() throws IOException
    {
        this.green = createSpot("green");
        this.green.setLight(new E27LightBulb());

        this.yellow = createSpot("yellow");
        this.yellow.setLight(new E27LightBulb());

        this.red = createSpot("red");
        this.red.setLight(new E27LightBulb());

        String path = this.currentRelativePath();
        URL url = this.getClass().getResource(path + "img/trafficLight.png");
        this.mask = ImageIO.read(url);
    }

    private void configurePositions()
    {
        final int WIDTH = (this.dimension.width - 20);
        final int HEIGHT = ((this.dimension.height - 30) / 3 );
        final Dimension dimension = new Dimension(WIDTH, HEIGHT);

```

```
76          int xLeft = this.position.x + 10;
77          int yTop = this.position.y + 10;
78          this.green.setPosition(xLeft, yTop);
79          this.green.setDimension(dimension);
80
81          yTop = (yTop + 5 + HEIGHT);
82          this.yellow.setPosition(xLeft, yTop);
83          this.yellow.setDimension(dimension);
84
85          yTop = (yTop + 5 + HEIGHT);
86          this.red.setPosition(xLeft, yTop);
87          this.red.setDimension(dimension);
88      }
89
90      public SimpleTrafficLight() throws IOException
91      {
92          this.create();
93          this.configurePositions();
94      }
95
96      public SimpleTrafficLight(Point position, Dimension dimension) throws IOException
97      {
98          this.position = position;
99          this.dimension = dimension;
100
101         this.create();
102         this.configurePositions();
103     }
104
105     public void setPosition(Point position)
106     {
107         this.position = new Point(position);
108         this.configurePositions();
109     }
110
111     public void setDimension(Dimension dimension)
112     {
113         this.dimension = new Dimension(dimension);
114         this.configurePositions();
115     }
116
117     public Point getPosition()
118     {
119         return new Point(this.position);
120     }
121
122     public Dimension getDimension()
123     {
124         return new Dimension(this.dimension);
125     }
126
127     @Override
128     public void paint(Graphics g)
129     {
130         synchronized (g)
131         {
132             int xLeft = this.position.x;
133             int yTop = this.position.y;
134             int width = this.dimension.width;
135             int height = this.dimension.height;
136
137             g.drawImage(mask, xLeft, yTop, width, height, null);
```

```
138
139             this.green.paint(g);
140             this.yellow.paint(g);
141             this.red.paint(g);
142         }
143     }
144
145     @Override
146     public TurnOnOff spotGreen() {
147         return this.spotGreen();
148     }
149
150     @Override
151     public TurnOnOff spotYellow() {
152         return this.spotYellow();
153     }
154
155     @Override
156     public TurnOnOff spotRed() {
157         return this.spotRed();
158     }
159
160
161 }
162
```